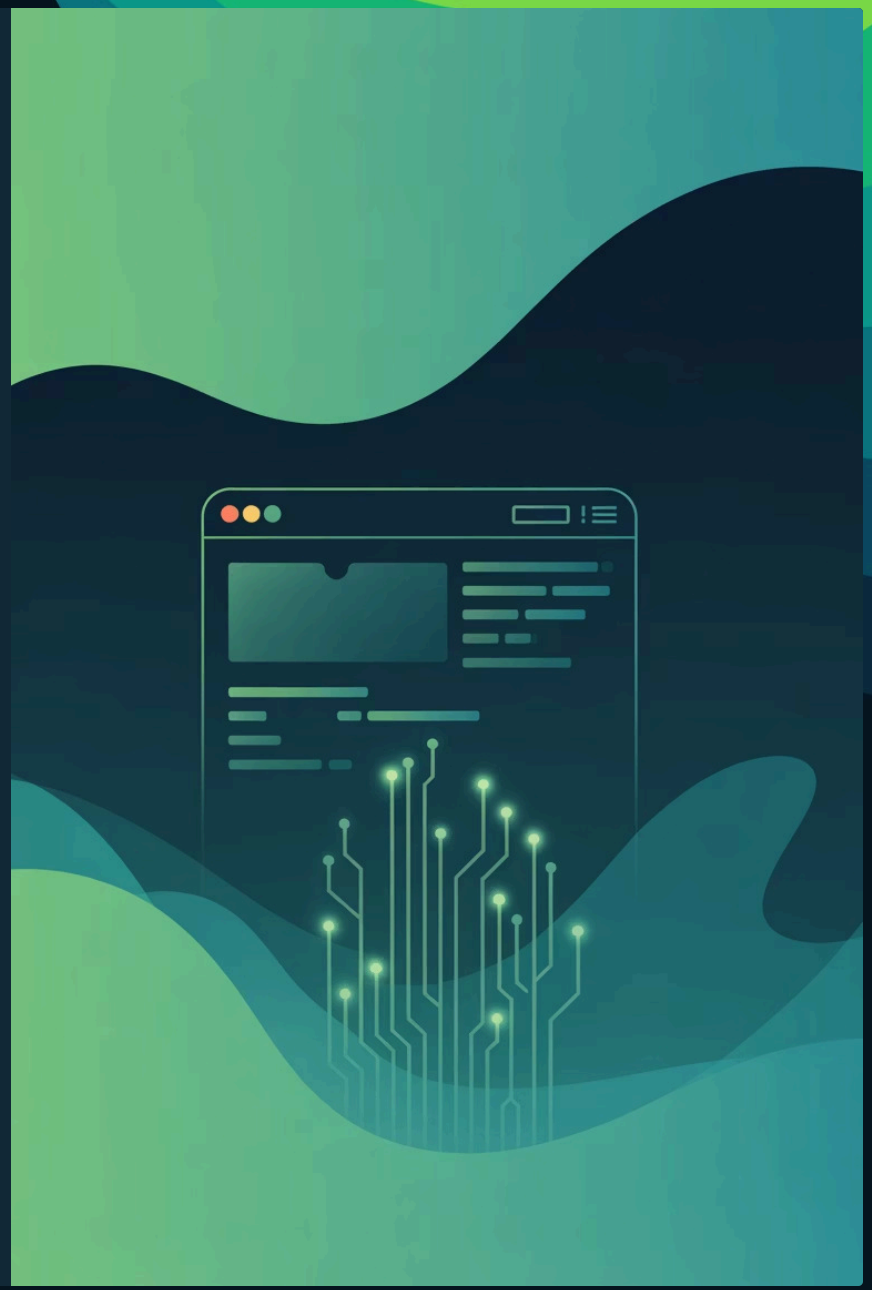
The Last Meter Interface

Web Control for Your Homebrew Hardware

Ilia Motornyi

Turku 2025



The DIY Project Predicament

DIY projects, from smart home gadgets to LED matrices, often hit a wall: building a user interface using and HID components a lot of code . This UI complexity frustrates makers, drains resources, and leaves ambitious hardware gathering dust, unfinished.

Lets use frontend

To control the device.

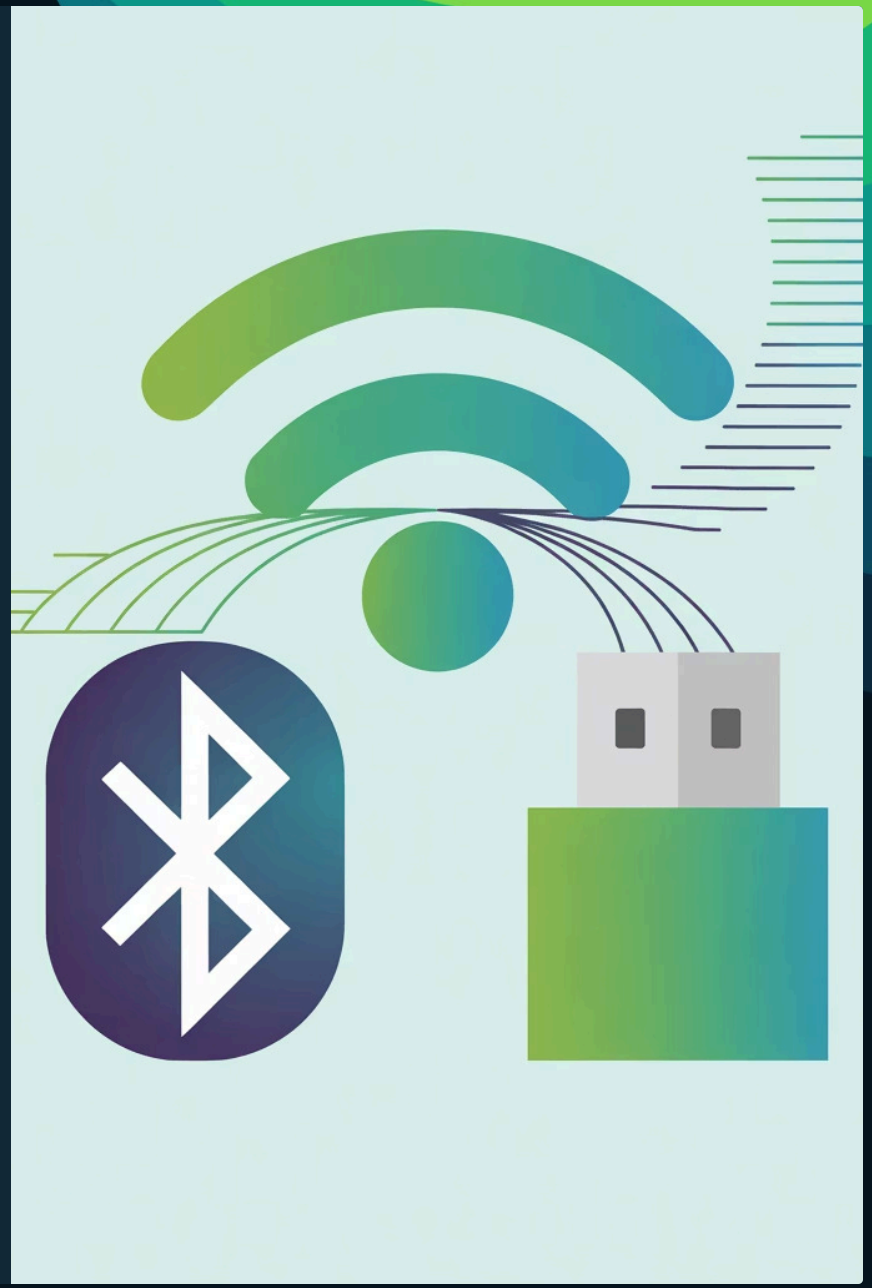
How to connect?

Wireless:

- Homelab server
- WiFi-connected MQTT client
- Bluetooth Classic Serial Port Profile via WebSerial
- Bluetooth Low Energy(BLE) via WebBluetooth

Wired:

- USB-UART via WebSerial
- USB Human Interface Device via WebHID USB
- Vendor-specific device via WebUSB



Homelab Server

Raspberry Pi

Pros:

- Familiar setup
- High performance
- A lot of disk and memory resources
- Does not depend on external services
- No browser limitations

Contras:

- Price
- Hardly be movable
- Can't be wearable
- Hardly be running on batteries
- Noticeable boot time
- Overengineering

Cloud MQTT

Pros:

- Lower power consumption
- Easy to implement
- Does not depend on external services
- No device or browser limitations

Contras:

- Requires WiFi-enabled device
- Does not work outside home
- Limited to be wearable
- Hardly be running on batteries
- High roundtrip latency

USB-UART and WebSerial

Pros:

- Easy protocol
- Even classic Arduino can be used
- Does not depend on external services

Contras:

- Does not work on Android or iOS
- Requires drivers on Windows
- Configuration hell
- Non-wearable

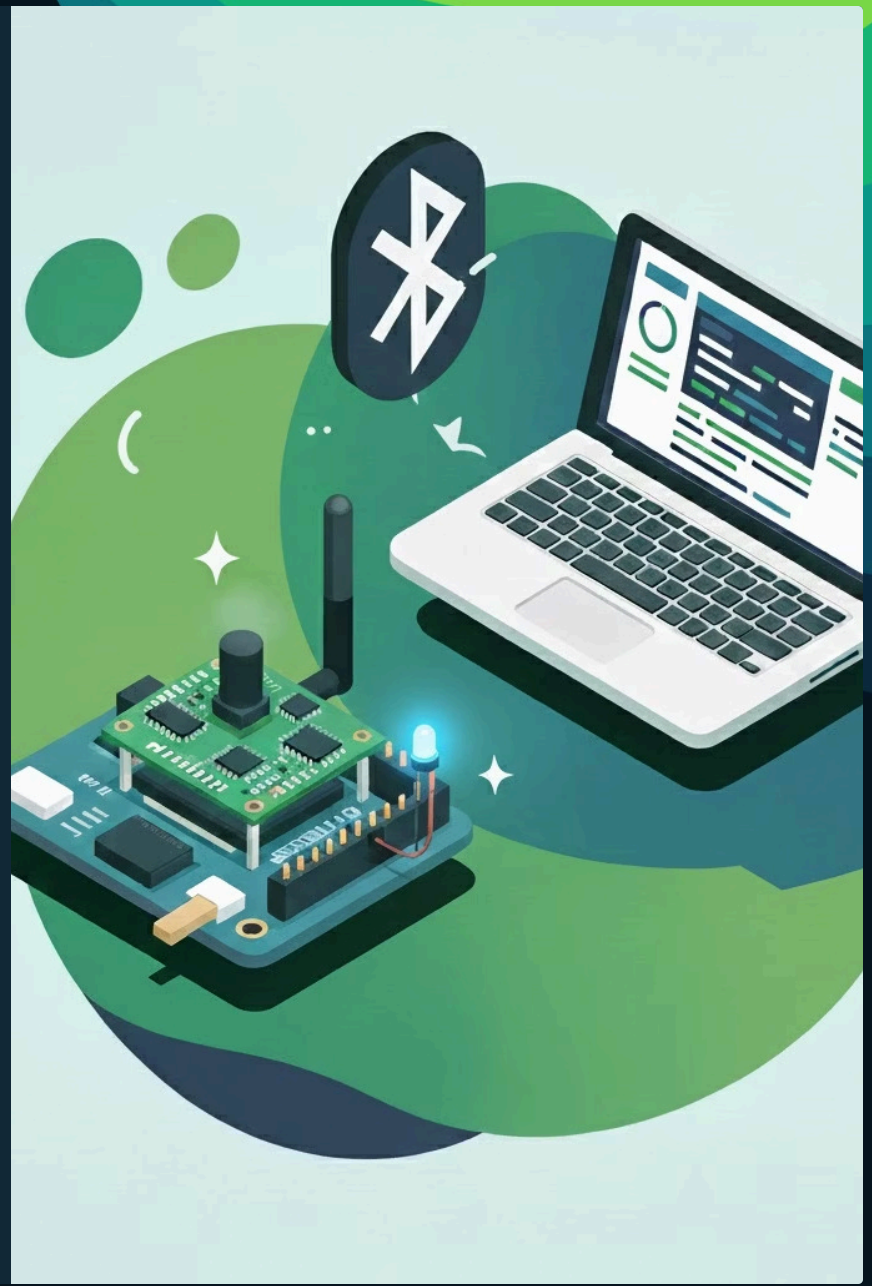
Bluetooth SPP and WebSerial

Pros:

- Easy protocol
- Wearable
- Does not depend on external services

Contras:

- Works only on Windows/Linux
- Configuration hell



BLE via WebBluetooth

Pros:

- Low power consumption
- Moderate complicated to implement
- Does not depend on external services
- Wearable

Contras:

- Requires BLE enabled device
- Does not connect to internet without gateways
- No iPhones

BLE GATT in a nutshell

for WebBluetooth programmers

The Generic Attribute Profile (GATT) defines how data is organized and exchanged over a Bluetooth Low Energy (BLE) connection

GATT Server ≈ Device

Hierarchical structure of services and characteristics

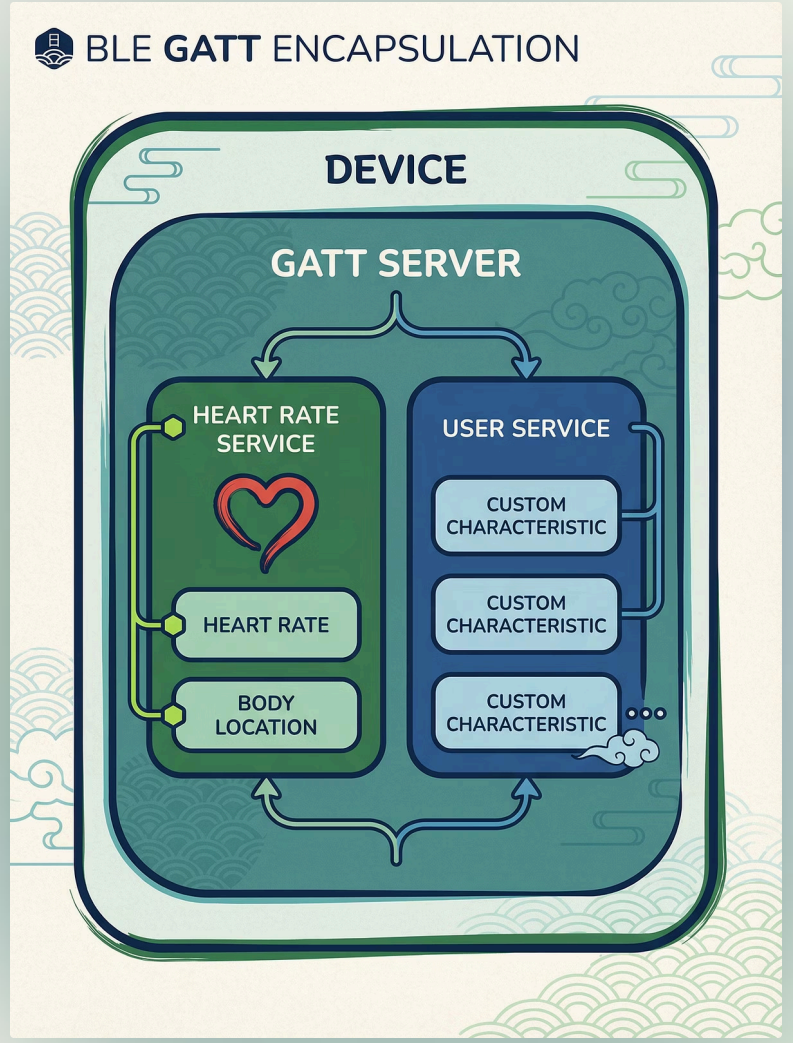
Service

A collection of variables (characteristics)

Examples include a "Heart Rate Service" or a "Battery Service."

Characteristic

A single data point within a service. It has a value and access mode (read, write, notify)



USB HID via WebHID

Pros:

- No extra power
- Does not depend on external services

Contras:

- Not supported iPhone & Android
- Non-wearable
- USB is complicated

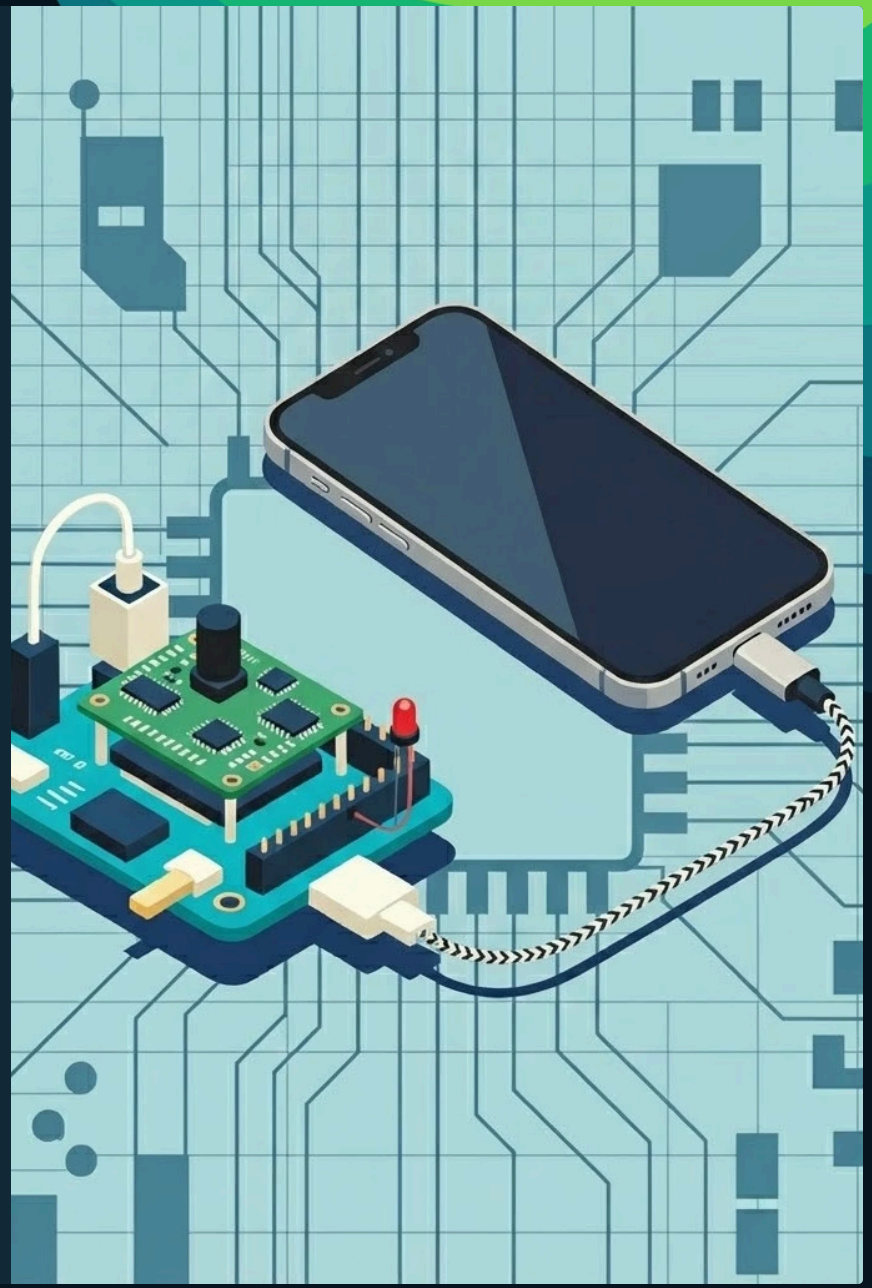
USB Vendor Specific Device via WebUSB

Pros:

- Does not depend on external services
- Can be attached to Android device

Contras:

- Complicated USB descriptor
- iPhone is not supported



USB in a nutshell

for WebUSB programmers

→ **Device identification**

Name

Vendor ID/ Product ID

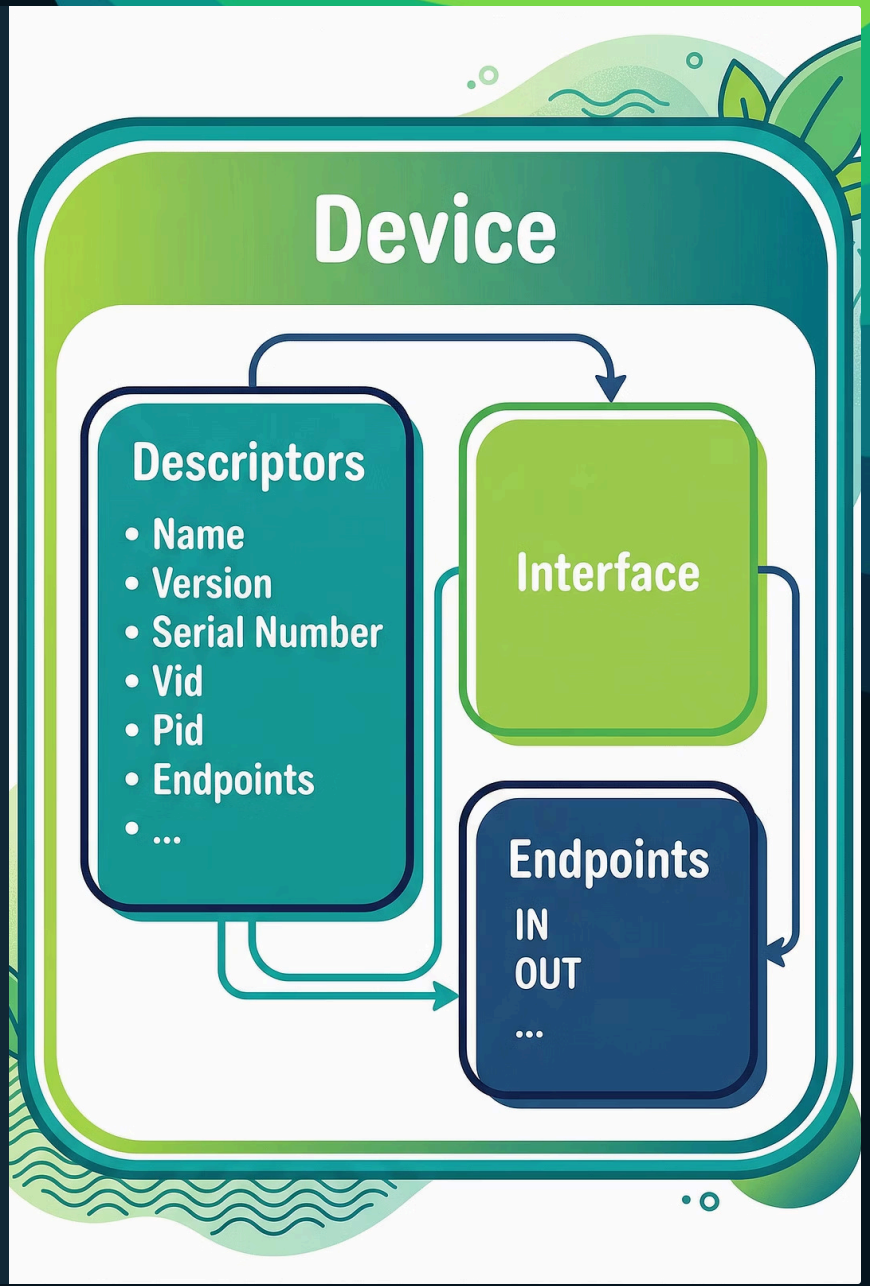
Serial number

→ **Descriptors**

Represents internal device functions to the USB Host

→ **Endpoints**

Data exchange



Thanks for watching

https://github.com/elmot/2025-speech-last-mile

https://linkedin.com/in/elmot/

/https://reddit.com/u/_elmot/